

CONTEXTUAL CLASSIFICATION ON THE MASSIVELY PARALLEL PROCESSOR

James C. Tilton
NASA Goddard Space Flight Center
Greenbelt, MD 20771, U.S.A.

ABSTRACT

Classifiers are often used to produce land cover maps from multispectral earth observation imagery. Conventionally, these classifiers have been designed to exploit the spectral (and, for multi-date data sets, temporal) information contained in the imagery. Very few classifiers exploit the spatial information content of the imagery, and the few that do rarely exploit spatial information content in conjunction with spectral and/or temporal information. We are studying a contextual classifier that exploits spatial and spectral information in combination through a general statistical approach. Early test results obtained from an implementation of the classifier on a VAX-11/780 minicomputer were encouraging, but they are of limited meaning because they were produced from small (50-by-50 pixel) data sets. Here we present an implementation of the contextual classifier on the Massively Parallel Processor (MPP) at the Goddard Space Flight Center (GSFC) that for the first time makes feasible the testing of the classifier on large data sets.

Keywords: Image classification, image pattern recognition, image contextual analysis, parallel processing, earth remote sensing.

INTRODUCTION

Algorithms that are currently used in most multispectral classification studies are unable to exploit the full spatial resolution of the Thematic Mapper (TM) data. Paradoxically, these algorithms often produce more accurate

classifications if the spatial resolution is degraded from 30 meters to the 80 meter resolution of Multispectral Scanner (MSS) data (Refs. 1,2), whereas humans can visually identify features more accurately in TM data at its original spatial resolution. This paradox is explained by noting that humans routinely use spatial information to help identify features in an image, while current commonly used classification algorithms do not use spatial information at all. The contextual classifier discussed here, however, does exploit spatial information, and has the potential of producing more accurate classifications of TM imagery at full resolution.

This contextual classifier was developed at Purdue University (Refs. 3,4), but it was tested only on 50-by-50 pixel data sets. The results produced in these tests were encouraging, but they were of limited value because of the small size of the test data sets. The classifier was not tested on larger data sets because it took too long to run on a VAX-11/780 minicomputer.

Testing the contextual classifier on large data sets becomes feasible when the algorithm is implemented on a massively (or fine-grained) parallel computer. Such a parallel computer is the Massively Parallel Processor (MPP) at the NASA Goddard Space Flight Center. The MPP is a Single Instruction, Multiple Data stream (SIMD) computer which was built by Goodyear Aerospace for the NASA Goddard Space Flight Center (Refs. 5,6). It consists of 16,384 bit serial microprocessors connected in a

128-by-128 mesh array with each element have data transfer connections with its four nearest neighbors. With this architecture, the MPP is capable of billions of operations per second.

A version of the contextual classifier has been implemented on the MPP, and a test of the classifier on the MPP took a total of 5 minutes to produce a 120-by-120 pixel classification. It would take roughly 12 hours to perform the same classification on a VAX-11/780 minicomputer. A 512-by-512 pixel classification takes one to two hours on the MPP (depending on parameter settings), whereas it would take one to two weeks to complete on a VAX-11/780 minicomputer. This more than a 100-fold improvement in running time has been obtained with a program written in a high level language on the MPP (MPP Pascal) with no concerted effort to optimize the program. We anticipate an additional 5 to 10-fold improvement in program running time with a highly optimized version of the program on the MPP.

We first present a derivation of the contextual classification decision rule, followed by a description of the implementation of the contextual classifier on the MPP. We close with some preliminary test results.

DERIVATION OF THE CONTEXTUAL CLASSIFICATION DECISION RULE

In the contextual approach to classification, the probable classifications of neighboring pixels influence the classification of each pixel. Classification accuracies can be improved through this approach since certain ground-cover classes naturally tend to occur more frequently in some contexts than in others. The contextual classifier that we have implemented on the MPP is the algorithm formulated by Swain et al (Ref. 3) and further developed by Tilton et al (Ref. 4). Here compound decision theory is invoked to develop a classification method which exploits spectral and

spatial information.

The derivation of the decision rule for the contextual classifier assumes that the data can be modeled as a two-dimensional array of $N = N_1 \times N_2$ of picture elements (pixels). At each pixel location (i,j) we are given an n -dimensional observation X_{ij} which is assumed to be a random sample from a distribution characteristic of the fixed but unknown true classification θ_{ij} . The observation X_{ij} usually contains spectral and/or temporal information about the pixel location (i,j) , and the classification θ_{ij} can be any one of m spectral or ground cover classes from the set $\Omega = \{\omega_i\}$, $i = 1, 2, \dots, m$.

In its most general form, the theory allows for a decision rule that is different for each pixel in the image, and, for each pixel, depends on the context of the entire image, $X = \{X_{ij} | i=1,2,\dots,N_1; j=1,2,\dots,N_2\}$. To obtain a tractable decision rule, however, we restrict the decision rule to be fixed for the entire image, and the context to be a subset of the entire image.

Define the context of the pixel at location (i,j) as $p-1$ observations spatially near, but not necessarily adjacent to, the observation X_{ij} . These $p-1$ contextual observations are taken from the same spatial positions relative to pixel position (i,j) for all i and j . Call this arrangement of pixels together with X_{ij} the p -context array. (A common p -context array for $p=5$ would be the observation X_{ij} at pixel (i,j) and the observations at the four nearest neighbor locations to pixel (i,j) .) Group the p observations in the p -context array into a vector of observations $X_{ij} = (X_1, X_2, \dots, X_p)^T$ and let θ_{ij} be the vector of true but unknown classifications associated with

the observation X_{ij} . Let $\theta^p \in \Omega^p$ and $X^p \in (R^n)^p$ stand respectively for p -dimensional vectors of classes and n -dimensional measurements; each component of θ^p is a variable which can take on any classification value $\Omega = \{\omega_i\}$, $i = 1, 2, \dots, m$; each component of X^p is a random n -dimensional vector which can take on values in the observation space. Correspondence of the components of X_{ij} , θ_{ij} , X^p , and θ^p to the positions in the p -context array is fixed but arbitrary, except that the p th component always corresponds to the pixel being classified.

We can now develop a decision rule, $d(X_{ij})$, which assigns a minimum risk classification to pixel (i,j) based on the vector of observations X_{ij} . The loss suffered by making the classification decision $d(X_{ij})$ for pixel (i,j) when the true class is θ_{ij} is denoted by $\lambda(\theta_{ij}, d(X_{ij}))$ for some fixed non-negative function $\lambda(\cdot, \cdot)$. The expected average loss (or risk) over the entire image is then

$$\begin{aligned} R_\theta &= E \left[\frac{1}{N} \sum_{i,j} \lambda(\theta_{ij}, d(X_{ij})) \right] \\ &= \frac{1}{N} \sum_{i,j} E[\lambda(\theta_{ij}, d(X_{ij}))] \\ &= \frac{1}{N} \sum_{\theta^p \in \Omega^p} \sum_{\substack{i,j \text{ with} \\ \theta_{ij} = \theta_p}} E[\lambda(\theta_p, d(X_{ij}))] \\ &= \sum_{\theta^p \in \Omega^p} \frac{1}{N} \sum_{\substack{i,j \text{ with} \\ \theta_{ij} = \theta_p}} \int \lambda(\theta_p, d(X^p)) f(X^p | \theta^p) dX^p \\ &= \sum_{\theta^p \in \Omega^p} G(\theta^p) \int \lambda(\theta_p, d(X^p)) f(X^p | \theta^p) dX^p \end{aligned}$$

$$= \int \sum_{\theta^p \in \Omega^p} G(\theta^p) \lambda(\theta_p, d(X^p)) f(X^p | \theta^p) dX^p \quad (1)$$

where θ_p is the p^{th} component of θ^p , and $G(\theta^p)$, the context function, is the relative frequency with which θ^p occurs in the array θ . For any array θ , a decision rule $d(X^p)$ minimizing R_θ can be obtained by minimizing the integrand of Equation 1 for each X^p ; thus for a specific X_{ij} (an instance of X^p), an optimal action is:

$d(X_{ij})$ = the action (classification) α which minimizes

$$\sum_{\theta^p \in \Omega^p} G(\theta^p) \lambda(\theta_p, \alpha) f(X_{ij} | \theta^p). \quad (2)$$

In practice, a "0-1 loss function" is employed, giving

$$\lambda(\theta, \alpha) = \begin{cases} 0, & \text{if } \theta = \alpha \\ 1, & \text{if } \theta \neq \alpha. \end{cases}$$

Then Equation 2 simplifies, and the decision rule becomes:

$d(X_{ij})$ = the action (classification) α which maximizes

$$\sum_{\substack{\theta^p \in \Omega^p, \\ \theta_p = \alpha}} G(\theta^p) f(X_{ij} | \theta^p). \quad (3)$$

A further assumption we make at this point is class-conditional independence of the observations (pixels) for any observation vector X_{ij} . In this case,

$$f(X_{ij} | \theta^p) = \prod_{k=1}^p f(X_k | \theta_k) \quad (4)$$

where X_k and θ_k are the k^{th} elements of X_{ij} and θ^p , respectively. Evidence that this is a reasonable assumption for Landsat MSS data may be found in Ref. 7. Invoking the class-conditional independence assumption, the decision rule (Equation 3) becomes:

$d(X_{ij})$ = the action (classification) α which maximizes

$$\sum_{\substack{\theta^p \in \Omega^p \\ \theta_p = \alpha}} G(\theta^p) \prod_{k=1}^p f(X_k | \theta_k). \quad (5)$$

Methods for estimating the context function $G(\theta^p)$ are discussed in Ref. 4. We use the "unbiased estimator", which is the most flexible and successful of these methods. Using this method, we first generate an unbiased estimate of a priori probabilities for each class at each position in the context array using the method described in Ref. 4. The product of these a priori probabilities is then calculated over the context array, forming the unbiased estimate of $G(\theta^p)$ based on one image point. The final estimate of $G(\theta^p)$ is made by averaging the individual point estimates over a portion of the data.

Conventional multispectral classifiers often classify into spectral classes (spectrally differentiable subclasses) rather than directly into the ground cover classes of interest. The spectral class classification is normally renumbered in a post-processing step to produce a classification map in terms of the ground cover classes. When the classification is done in terms of spectral classes, we assume that $f(X_k | \theta_k)$ is a multivariate normal density with mean vector and covariance matrix determined by the class, θ_k . In the case where the classification is done in terms of ground cover classes,

we assume that $f(X_k | \theta_k)$ is a weighted sum of multivariate normal densities, viz.

$$f(X_k | \theta_k) = \sum_{\zeta_k \in \theta_k} r(\zeta_k | \theta_k) g(X_k | \zeta_k) \quad (6)$$

where ζ_k is the k^{th} spectral class, $r(\zeta_k | \theta_k)$ is the conditional probability of spectral class ζ_k given ground cover class θ_k and $g(X_k | \zeta_k)$ is a multivariate normal density with mean vector and covariance matrix determined by the spectral class, ζ_k .

IMPLEMENTATION OF THE CONTEXTUAL CLASSIFIER ON THE MPP

In both the parallel MPP implementation, and the conventional serial implementation, classification directly into ground cover classes generally requires significantly less computer time than a classification into spectral classes (Ref. 4). Let m be the number of ground cover classes, c be the number of spectral classes ($c \geq m$), and p be the number of pixels in the p -context array. If, for example, $c=2m$, a contextual classification into spectral classes would have to consider $(2m)^p$ context configurations, while a contextual classification directly into ground cover classes would only have to consider m^p context configurations. If the classification is performed using four nearest neighbor context (i.e., $p=5$), then the spectral class classification would pass through the main loop in the contextual classification program a (multiplicative) factor of 32 times the number of passes that would be required for a ground cover class classification. Since the ratio of spectral classes to ground cover classes is often greater than 1.5 or so, we normally classify directly into ground cover classes with the contextual classifier.

Since the training classes are nearly always given as a set of multivariate normal distributions corresponding to spectral classes (in this case, the $g(X_k|\zeta_k)$ in Equation 6), we must first estimate the $r(\zeta_k|\theta_k)$ in Equation 6 in order to calculate the $f(X_k|\theta_k)$ used in the contextual classification decision rule, Equation 5. In our implementation, the same unbiased estimator used to estimate the a priori probabilities for the context function is used to estimate the $r(\zeta_k|\theta_k)$ by limiting the classes ζ_k to the spectral classes associated with ground cover class θ_k . This step can be considered to be a preprocessing step, and is in fact implemented as a separate MPP program. In our implementation, we use the MPP to calculate the average value of $g(X_k|\zeta_k)$ for each ζ_k over the entire data set (the program cycles through as many 128-by-128 pixel sections of data as required to cover the entire data set), and return to the host VAX-11/780 minicomputer to do the remaining serial calculations required to compute the estimate of the $r(\zeta_k|\theta_k)$.

The MPP implementation of the main portion of the contextual classifier has several advantages over a conventional serial implementation. The obvious advantage is that calculations for 16384 pixels can be performed in parallel. Less obviously, there are further algorithmic advantages to an MPP implementation. The MPP parallel architecture makes it possible to estimate the context function, $G(\theta^p)$, and perform the summation in the decision rule (Equation 5) in one pass through the data. In a serial implementation, the context function $G(\theta^p)$ must be estimated in one pass through a portion of data, and the decision rule must then be evaluated in a second pass. This implementation feature gives a clear efficiency advantage to the MPP implementation. In addition, this

feature also gives a subtle accuracy advantage to the MPP implementation since now we can obtain unique estimates of the context function for each pixel. In fact, with the MPP parallel architecture it actually costs less to compute unique values of the context function for each pixel than to compute a block average value of the context function. Because of computation and core memory limitations, a serial implementation is forced to use one average estimate of the context function in classifying a block of data (in Ref. 4 the block sizes ranged from 10-by-10 to 25-by-25 pixels).

Now we describe the MPP implementation of the contextual classifier in more detail. (For a detailed description of the serial implementation see Ref. 4.) Since the MPP consists of an array of 128-by-128 microprocessors, the contextual classification is performed on 128-by-128 pixel portions of multispectral data. To classify an entire data set, 128-by-128 pixel portions of data must be cycled through the program. (These portions of data must overlap by a certain number of pixels determined by the area over which the context function is estimated -- see below.)

Before the program's main classification loop is entered, the class-conditional probabilities, $f(X_k|\theta_k)$, are calculated for each pixel, and an unbiased estimate of the a priori probabilities of each class is made for each pixel. The main classification loop consists of an outside loop over the ground cover classes ' α ' and an inside loop over all possible classification vectors θ^p with $\theta_p = \alpha$ (see Equation 5).

Inside the main classification loop, the context function is estimated for the given combination of classes in the context array. A unique estimate of the context function for each pixel is made from an N-by-N square of data centered at each pixel (typically $9 \leq N$

(≤ 25). The estimate for pixels on the outer $N/2$ pixel edge of the array is taken to be zero and no classification is performed for those pixels. Then the product is formed between the context function value at each pixel and the class-conditional probabilities across the context array giving the contribution to the discriminant function for the given combination of classes. The discriminant function for ground cover class ' α ' is accumulated by continuing the loop through all possible classification vectors θ^p with $\theta_p = \alpha$. Once the discriminant functions have been calculated for all ground cover classes, the classification result at each pixel is taken to be the class with the maximum discriminant function at that pixel.

The direct implementation of the contextual classification decision rule (Equation 5) on either a serial (e.g. VAX-11/780) or parallel (e.g. MPP) computer runs into a problem of insufficient exponential range on most computers. For example, and both the MPP and VAX-11/780 computers, the magnitude range of single precision floating point numbers is approximately $0.29e-38$ to $1.7e+38$. (Due to efficiency considerations and that fact the MPP currently has no double precision floating point implemented, we do not consider double precision floating point numbers here.) With four-nearest-neighbor context ($p=5$), we see from the previous paragraph that the estimation of the context function, $G(\theta^p)$, involves the multiplication of 5 numbers. Thus Equation 5 requires the multiplication of a total of 10 numbers together. Since each of these numbers must lie in the range $0.0 < 1.0$, and, in practice, often lie in the range $0.0 < 1.0e-4$, it is easy to underflow the decision rule and be unable to determine a classification for many image pixels. This difficulty is dealt with by evaluating the natural logarithm (LN) of the decision rule rather than the decision rule directly. This trick effectively compresses the

exponential range. For example, an exponential range of $1.0e+38$ to $1.0e-38$ is compressed to the range of numbers $+87.5$ to -87.5 . (This trick does cause a loss of precision, which, however, is of no consequence here.)

Let

$$d_{\alpha}(X_{ij}) = \sum_{\substack{\theta^p \in \Omega^p \\ \theta_p = \alpha}} G(\theta^p) \prod_{k=1}^p f(X_k | \theta_k) \quad (7)$$

and

$$d'_{\alpha}(X_{ij}) = \text{LN}(d_{\alpha}(X_{ij})). \quad (8)$$

Maximization of $d_{\alpha}(X_{ij})$ in Equation 5 (and 7) based on $d'_{\alpha}(X_{ij})$ is equivalent to maximization based on $d_{\alpha}(X_{ij})$. Thus, the decision rule becomes:

$d(X_{ij})$ = the action (classification) α which maximizes

$$d'_{\alpha}(X_{ij}) = \text{LN} \left(\sum_{\substack{\theta^p \in \Omega^p \\ \theta_p = \alpha}} G(\theta^p) \prod_{k=1}^p f(X_k | \theta_k) \right). \quad (9)$$

Let

$$F(X_{ij}, \theta^p) = \text{LN} \left(G(\theta^p) \prod_{k=1}^p f(X_k | \theta_k) \right) \quad (10)$$

and

$$M_{\alpha}(X_{ij}) = \text{MAX}_{\substack{\theta^p \in \Omega^p \\ \theta_p = \alpha}} \left(F(X_{ij}, \theta^p) \right). \quad (11)$$

Then

$$\begin{aligned}
 d'_\alpha(\mathbf{X}_{ij}) &= \text{LN} \left(\sum_{\theta_p = \alpha} \text{EXP}[F(\mathbf{X}_{ij}, \theta^p)] \right) \\
 &= \text{LN} \left(\sum_{\theta_p = \alpha} \text{EXP}[F(\mathbf{X}_{ij}, \theta^p) - M_\alpha(\mathbf{X}_{ij}) + M_\alpha(\mathbf{X}_{ij})] \right) \\
 &= \text{LN} \left(\text{EXP}[M_\alpha(\mathbf{X}_{ij})] \sum_{\theta_p = \alpha} \text{EXP}[F(\mathbf{X}_{ij}, \theta^p) - M_\alpha(\mathbf{X}_{ij})] \right) \\
 &= M_\alpha(\mathbf{X}_{ij}) + \text{LN} \left(\sum_{\theta_p = \alpha} \text{EXP}[F(\mathbf{X}_{ij}, \theta^p) - M_\alpha(\mathbf{X}_{ij})] \right)
 \end{aligned} \tag{12}$$

Calculating $d'_\alpha(\mathbf{X}_{ij})$ in this way insures that at least one term of the sum does not cause underflow, because the exponent of the maximum term, $M_\alpha(\mathbf{X}_{ij})$, is never taken. This procedure also makes it less likely that other terms in the sum will underflow since the $F(\mathbf{X}_{ij}, \theta^p)$ tend to be large **negative** numbers.

Note that Equation 10 can be rewritten as:

$$F(\mathbf{X}_{ij}, \theta^p) = \text{LN}[G(\theta^p)] + \sum_{k=1}^p \text{LN}[f(\mathbf{X}_k | \theta_k)] \tag{13}$$

When evaluated in this way $F(\mathbf{X}_{ij}, \theta^p)$, and thus $d'_\alpha(\mathbf{X}_{ij})$, do not require any

multiplications. All multiplications are replaced by sums of natural logarithms of the terms.

The value of $M_\alpha(\mathbf{X}_{ij})$ is not known prior to the start of the summation in Equation 12. Theoretically we could use the maximum value of $F(\mathbf{X}_{ij}, \theta^p)$ found up to the current term of the sum, and reshuffle the terms of Equation 12 when a new maximum is found. However, the limits of the exponential range on the MPP (approx. $1.0\text{E}+-38$) make the use of this technique impractical (an implementation "trick" along these lines may still be pursued, however).

The current implementation of the contextual classifier executes a loop over the $\theta^p \in \Omega^p$ once to identify the value of $M_\alpha(\mathbf{X}_{ij})$, and actually evaluates Equation 12 in a second execution of the loop. We have noticed

previously in Reference 8, however, that the following decision function produces classifications that closely approximate those produced by the decision function in Equation 12:

$$d(X_{ij}) = \text{the action } \alpha \text{ which} \\ \text{maximizes } M_{\alpha}(X_{ij}), \quad (14a)$$

or in the notation of Equation 9:

$d(X_{ij})$ = the action α which maximizes
for all $\theta^p \in \Omega^p$ with $\theta_p = \alpha$

$$d'_{\alpha}(X_{ij}) = \text{LN} \left[G(\theta^p) \prod_{k=1}^p f(X_k | \theta_k) \right]. \quad (14b)$$

This approximate version of the contextual classifier is also implemented on the MPP. The advantage of approximate version is that the the loop over the $\theta^p \in \Omega^p$ need be performed only once.

One more implementation comment is relevant here. Running on the MPP host VAX-11/780 minicomputer is the Land Analysis System (LAS), a package of numerous image analysis and manipulation programs. The LAS is implemented under the Transportable Applications Executive (TAE), which is a portable, uniform, user-friendly user interface. Since we eventually want to make the Contextual Classifier available to researchers from a wide range of earth science applications, we have implemented the Contextual Classifier under TAE and made all image and data files conform to LAS standards.

PRELIMINARY CONTEXTUAL CLASSIFICATION RESULTS

We have thus far obtained preliminary contextual classification results on two data sets using the MPP

implementation of the contextual classifier. Other results using a VAX-11/780 minicomputer implementation of the contextual classifier are given in References 3 and 4.

The first data set we will discuss is a subset of a Landsat Thematic Mapper image from northern Virginia near the town of Bowling Green. The data set was developed originally for another study (Ref. 9). This area includes Fort A. P. Hill for which there is extensive ground truth data. (However, only 271 pixels of ground truth data have been extracted and registered for accuracy assessment. A more complete extraction of ground truth data from the air photography is being considered.) Being located only 50 miles south of Washington, D. C., the study area was readily accessible for field investigation to the confirm ground truth data.

According to the investigators who originally developed this data set, "the topography of this part of Virginia consists of gently rolling hills with agricultural areas along the flood plains, marsh and swamps in low lying areas adjacent to rivers and streams, and forests in the upland. The Rappahannock River runs across the northern portion of the study area and there are a number of streams that drain into it. The main types of vegetation in the area are deciduous and coniferous trees, marsh and pasture grasses, and an assortment of agricultural crops. The principal agricultural crops grown here are corn, soybean, and wheat" (Ref. 9).

The version of the data set used in our study is described in the original study as the "full resolution combined dates (full comb.)" data set. This data set consists of registered multi-date 30 meter resolution Thematic Mapper data from March 5, 1984; July 29, 1982; and November 2, 1982. Bands 3, 4 and 5 of the March and November data sets were used and bands 3 and 4 of the July data set was used. We did not develop our own multivariate normal

model for the ground cover classes in the scene, but instead used the mean vectors and covariance matrices generated by the original study for our class-conditional density functions. These classes were obtained through a supervised technique resulting in covariance matrices with generally much less spread than covariance matrices obtained from the common unsupervised clustering technique for generating the class-conditional density functions.

(This data set was used to shake-down the implementation of the algorithms. We encountered some difficulty in our early implementation of the algorithms due to the fact that the covariance matrices had very little spread. Because of this, the entire data set was not truly represented by the classes chosen and some data points produced low values for all class-conditional density functions. We found that simple thresholding was not satisfactory, and had to include normalization steps in the implementation of the unbiased estimator. This was all complicated by the fact that we implemented the algorithms on the NASA/Goddard Massively Parallel Processor which for a time had floating-point math without underflow and overflow detection. We had to wait for an implementation of underflow detection before the algorithm worked properly. Underflow detection may not have been required for covariance matrices with wider spreads.)

For this data set we obtained an overall classification accuracy of 79.7% (216 correct classifications out of 271 test pixels) for the contextual classifier. This compares to an overall classification accuracy of 77.5% (210 correct classifications out of 271 test pixels) for a conventional per-pixel uniform-priors maximum likelihood classification. This conventional classification was obtained using the standard BAYES classification program in the Goddard Land Analysis System (LAS) software package. We evaluated over five ground

cover classes: wetlands (and seasonal wetlands), water, barren land, forest and agriculture. The full classification contains 158,105 pixels (roughly 512 by 309 pixels), and was performed in less than one hour (wall clock time) on the MPP.

As mentioned earlier, the ground truth used for deriving the classification accuracy results for this data set consisted of manual ground cover class determinations at 271 pixel locations scattered throughout the data set (see Ref. 9). We feel that a better evaluation of the contextual classifier would be obtained by evaluating the classification results against a more extensive ground truth map. We are pursuing an effort to develop a more extensive ground truth map for the area from aerial photographs that were taken over the same time period when the TM data was gathered.

The next data set that we will discuss in the Anderson River airborne Multispectral Scanner (ABMSS) data set. This data set is a part of a SAR/MSS data set that was acquired, preprocessed, and loaned to us by the Canada Centre for Remote Sensing (CCRS), Department of Energy, Mines, and Resources, of the Government of Canada. This data set covers a 2.8km by 2.8km area in British Columbia, Canada near the Anderson River with terrain elevations ranging from 330 to 1100 meters above sea level. The data were geometrically corrected by CCRS to the Universal Transverse Mercator (UTM) projection at a spatial resolution of 50 meters. A pixel-by-pixel ground cover map was digitized by CCRS from a detailed forest cover map prepared by the staff of the Pacific Forest Research Centre of Canada from aerial photography and more than 20 ground plots (Reference 10).

For this data set we obtained an overall classification accuracy of 81.0% for the contextual classifier. This compares to an overall classification accuracy of 80.5% for the standard BAYES classification

program. We evaluated over three ground cover classes: clearcut, hemlock and douglas fir mix. The full data set is 57 pixels by 57 pixels of which the center 49 pixels by 49 pixels were classified by the contextual classifier (a four pixel border was required because of the 9-by-9 pixel window used to estimate the context function). Both the contextual classifier and the BAYES classifier were evaluated over the center 49-by-49 pixel portion of the ground truth data.

We are not happy with the class mean vectors and covariance matrices that we developed for this data set, especially since the original study of this data set obtained an overall accuracy of 88% using per-pixel classification techniques (Ref. 10). This result was obtained for a more difficult discrimination task of classifying into eight ground cover classes: douglas fir, douglas fir mixed with lodgepole pine, douglas fir mixed with cedar, douglas fir mixed with hemlock, hemlock mixed with douglas fir, hemlock mixed with cedar, clearcuts, and bare rock. We have contacted the Principal Investigator for the original study, and have arranged for obtaining the class mean vectors and covariance matrices that were developed for that study. Unfortunately, the publication schedule precludes including results using those class means and covariances in this paper.

CONCLUDING REMARKS

Earlier studies (Refs. 3 and 4) using a VAX-11/780 minicomputer implementation of the contextual classifier obtained classification accuracy improvements of 2% to nearly 6% for small 50-by-50 pixel data sets. These classification runs generally took 3 to 4 hours (wall-clock) to complete. We have implemented the contextual classifier on NASA Goddard's Massively Parallel Processor in order to enable the testing of the contextual classifier on reasonably sized data sets (e.g. 512-by-512 pixels).

Preliminary tests have shown that a 512-by-390 pixel data set can be classified with the contextual classifier in approximately one hour (wall-clock) on the MPP. In this implementation of the contextual classifier on the MPP we made no concerted effort to come up with the most efficient implementation possible on the MPP. Still, this relatively inefficient implementation provides better than a 100-fold speed-up over a fairly efficient VAX-11/780 implementation of the algorithm. This amount of speed-up is sufficient to make it possible for the first time to study the effectiveness of this classifier on several different data sets of reasonable size (e.g. 512-by-512 pixels).

The preliminary classification accuracy results reported in this paper for the MPP implementation of the contextual classifier are not as impressive as earlier results obtained from a VAX-11/780 minicomputer implementation of the classifier. Different data sets were used in the earlier study. Also, we expect that our results will improve once certain aforementioned problems are taken care of concerning the data sets used, and once the contextual classifier is run on several other well constructed data sets.

One final note. It makes little sense to compare the speed of the contextual classifier as implemented on a vector supercomputer such as a Cray to the speed of the implementation on the MPP. Devising an implementation on the MPP that effectively uses the parallelism of the MPP is very easy and natural, whereas it would be much more difficult to develop an implementation on a vector supercomputer that effectively exploits that type of parallelism. Being such an easy and natural implementation, the MPP implementation lends itself much more effectively to experimentation with the algorithm.

REFERENCES

1. Williams, D.L., J.R. Irons, B.L. Markham, R.F. Nelson, D.L. Toll, R.S. Latty, and M.L. Stauffer, "A Statistical Evaluation of the Advantages of LANDSAT Thematic Mapper Data in Comparison to Multispectral Scanner Data," *IEEE Trans. on Geoscience and Remote Sensing*, 1984, Vol. GE-22, No. 3, pp. 294-302.
2. Toll, D.L., "Effect of Landsat Thematic Mapper Sensor Parameters on Land Cover Classification," *Remote Sensing of Environment*, 1985, Vol. 17, pp. 129-140.
3. Swain, P.H., S.B. Vardeman, and J.C. Tilton, "Contextual Classification of Multispectral Image Data," *Pattern Recognition*, 1981, Vol. 13, No. 6, pp. 429-441.
4. Tilton, J.C., S.B. Vardeman, and P.H. Swain, "Estimation of Context for Statistical Classification of Multispectral Image Data," *IEEE Trans. on Geoscience and Remote Sensing*, 1982, Vol. GE-20, No. 4, pp. 445-452.
5. Batcher, K.E., "Design of a Massively Parallel Processor," *IEEE Trans. on Computers*, 1980, Vol. C-29, No. 9, pp. 836-840.
6. Schaefer, D.H., J.R. Fischer and K.R. Wallgren, "The Massively Parallel Processor," *AIAA Journal of Guidance, Control, and Dynamics*, 1982, Vol. 5, No. 3, pp. 313-315.
7. Kettig, R.L. and D.A. Landgrebe, "Computer Classification of Remotely Sensed Multispectral Image Data by Extraction and Classification of Homogeneous Objects," *LARS Technical Report 050975*, Laboratory for Applications of Remote Sensing (LARS), Purdue University, West Lafayette, Indiana 47907, May 1975.
8. Tilton, J.C., "Contextual Classification of Multispectral Image Data: Approximate Algorithm," *LARS Technical Report 081580*, Laboratory for Applications of Remote Sensing (LARS), Purdue University, West Lafayette, Indiana 47907, August 1980.
9. M. W. Snyder and M. H. Story, "Classification of Multispectral Thematic Mapper Data," *Technical Papers of the 1986 ACSM-ASPRS Annual Convention, Volume 4: Photogrammetry*, Washington, D. C., pp. 360-369, March 16-21, 1986.
10. D. G. Goodenough, B. Guidon, P. M. Teillet, A. Ménard, and J. Zelek, "The CCRS SAR/MSS Anderson River Data Set," presented at the *International Association on Pattern Recognition Technical Committee 7 Workshop*, Paris, France, October 23-24, 1986.